VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

METODY DETEKCE ZNAKOVÉ ŘEČI - REŠERŠNÍ STUDIE

SIGN LANGUAGE DETECTION METHODS - REVIEW

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — **LUBOŠ PETR**

AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. JIŘÍ KREJSA, Ph.D.**

SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	**Luboš Petr**
Studijní program:	Strojírenství
Studijní obor:	Základy strojního inženýrství
Vedoucí práce:	**doc. Ing. Jiří Krejsa, Ph.D.**
Akademický rok:	2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Metody detekce znakové řeči – rešeršní studie

Stručná charakteristika problematiky úkolu:

S pokrokem v oblasti senzoriky a metod zpracování signálu a obrazu je možné zpracovávat stále složitější úlohy detekce polohy člověka. Jednou z možných aplikací je detekce znakové řeči. Cílem práce je vytvořit ucelený souhrn metod používaných na tuto úlohu v současné době, spolu s přihlédnutím ke specifikům českého znakového jazyka.

Cíle bakalářské práce:

1. Vytvořte přehled metod používaných při detekci znakové řeči
2. Zhodnoťte metody z hlediska hardwarových a výpočetních nároků
3. Zhodnoťte metody z hlediska univerzálnosti použitého jazyka, s důrazem na češtinu.

Seznam doporučené literatury:

COOPER Helen, HOLT Brian, BOWDEN Richard: Sign Language Recognition, Chapter in Visual Analysis of Humans: Looking at People, Springer, pp. 539 - 562, 2011

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

Abstrakt
Cílem této práce je popsat různé metody detekce znakové řeči. Výstupem jednotlivých metod je funkční překlad znakové řeči do textu v reálném čase. Kromě detekce pomocí rukavic a zařízení kinect se tato práce zabývá možnostmi detekce znakové řeči z obrazového záznamu, což je vzhledem k dostupnosti do budoucna nejvíce perspektivní způsob detekce. Práce je dále zaměřena na klasifikaci znaků pomocí neuronových sítí.

Summary
The Aim of this work is to describe various methods of sign language detection. The output of individual methods is a functional translation of sign language into text in real time. In addition to glove and kinect detection, this work deals with the possibilities of sign language detection from image recording, which is the most prospective method of detection in the future. The thesis is also focused on sign classification using neural networks.

Klíčová slova
Znakový jazyk, detekce, kinect, neuronové sítě

Keywords
Sign language, detection, kinect, neural networks

PETR, L.*Metody detekce znakové řeči - rešeršní studie*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2019. 35 s. Vedoucí diplomové práce doc. Ing. Jiří Krejsa, Ph.D.

Prohlašuji, že svou bakalářskou práci na téma „Metody detekce znakové řeči - rešeršní studie“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce

Luboš Petr

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Jiřímu Krejsovi, Ph.D. za odborné vedení a konzultace. Dále bych chtěl poděkovat celé své rodině za podporu při studiích.

Luboš Petr

Obsah

1. Úvod

Znakový jazyk je prostředkem komunikace pro komunity neslyšících po celém světě. V České republice se vyskytuje zhruba 7600 neslyšících lidí používajících znakový jazyk. Ve světě se číslo sluchově postižených lidí využívající znakový jazyk blíží k 70 milionům.[1] Tato nemalá část populace má omezené prostředky ke komunikaci s ostatními lidmi. Návrh funkčního překladače znakového jazyka by byl velkým přínosem a zkvalitněním života pro většinu neslyšících. Ve světě existuje přes 300 různých znakových jazyků. K detekci každého z nich je potřeba vytvořit databázi znaků, která bude sloužit jako vstup do softwaru pro její klasifikaci.

Systémy detekující znakový jazyk se rozdělují do dvou kategorií: systémy využívající hardware a systémy využívající vizuální vstupy. Systémy využívající hardware vyžadují, aby uživatel nosil na rukách jistý typ senzorického zařízení například rukavice cyber glove. Tato zařízení měří pohyby a orientace ruky a prstů a v minulosti byla často využívána k detekci znakového jazyka. Systémy využívající vizuální vstupy zpracovávají obrazový záznam uživatele a pomocí klasifikačních metod hodnotí obsah záznamu znakující osoby. Tyto systémy nijak neomezují pohyb znakující osoby na rozdíl od systémů, které využívají rukavice, které do jisté míry omezují znakovací možnosti. Systémy zpracovávající obraz na druhou stranu mají vyšší výpočetní nároky.

Tato práce je zaměřena na popis hardwarových systémů využívajících rukavice, systémů využívajících zařízení Kinect, tak i systémů zpracovávajících obraz. Metodám zpracovávající obraz je v této práci věnována větší pozornost, jelikož jsou uživatelsky více přátelské.

2. Český znakový jazyk

2.1. Terminologie

V českém prostředí je nutné rozlišovat tři rozdílné termíny, které se v souvislosti s dorozumíváním s neslyšícími používají. Jsou to znaková řeč, znakovaná čeština a český znakový jazyk.

Znakovaná čeština je umělý systém, který byl vytvořen slyšícími k usnadnění způsobu komunikace s neslyšícími. Kopíruje gramatiku českého jazyka a používají ho většinou lidé ohluchlí nebo nedoslýchavý. Z těchto důvodů se nejedná o přirozený jazyk.

Český znakový jazyk je přirozený a plnohodnotný jazyk neslyšících s vlastní gramatikou a slovní zásobou. V neslyšící komunitě se používá mnohem častěji než znakovaná čeština.

Znaková řeč je termín, který je podle Zákona č. 155/1998 Sb. nadřazeným pojmem pro český znakový jazyk a znakovanou češtinu. Znaková řeč je v něm vymezena jako souhrnný, zastřešující pojem pro „vizuálně-motorické symbolické komunikační systémy“. Termín tak označuje něco co je vzájemně naprosto odlišné. Je ale rozšířený mezi většinovou společnost.[2]

V této práci se místo termínu znaková řeč budu držet přesnějšího pojmu Český znakový jazyk (dále ČZJ). Znakovaná čeština není vhodná, protože se v komunitě neslyšících příliš nepoužívá.

2.2. ČZJ jako přirozený jazyk

Český znakový jazyk patří stejně jako čeština a všechny ostatní znakové jazyky mezi jazyky přirozené. Přirozeným jazykem se rozumí jazyk, který vznikl přirozeným vývojem a je používaný při běžné komunikaci.[3] Pro přirozený jazyk platí základní kritéria a to dvojí artikulace, znakovost, svébytnost, systémovost, produktivnost a historický rozměr.

Dvojí artikulace má význam takový, že nejmenší jazykové jednotky nesoucí význam lze dále dělit na jednotky, které význam nenesou, ale můžou ho od sebe odlišit. V jazyce mluveném mluvíme o morfémech a fonémech, ve znakovém jazyce pak o znacích a fonémech.

Znakovost znamená, že díky systému znaků a jednotek, který má každý jazyk, lze nahradit reálnou věc znakem nebo slovem, aniž bychom si předmět museli reálně opatřit a ukazovat na něj.

Svébytnost je úzce spjatá se znakovostí. Díky svébytnosti dokážeme v jazyce vyjádřit podmínku, minulý nebo budoucí děj, a to díky tomu že všechna jazyková sdělení jsou nezávislá na reálné situaci.

Systémovost znamená, že je jazyk složen ze systému jednotek a pravidel která ho spojují. Toto kritérium naplňuje jak ČZJ tak i čeština.[4]

2.3. Hlavní rozdíly mezi českým jazykem a ČZJ

Hlavním rozdílem je samotná podstata obou jazyků. Čeština je jazyk mluvený a je stejně jako všechny mluvené jazyky jazykem audio-orálním. To znamená, že se nesená informace

šíří zvukem a je zpracována sluchovým ústrojím osoby, která informaci přijímá. ČZJ je jazykem vizuálně motorickým. Je tvořen pohyby rukou, těla, úst a mimikou mluvčího a je přijímán zrakem osoby přijímající informaci. Oba jazyky jsou omezeny množstvím zvuků nebo fónů. ČZJ je omezen také znakovacím prostorem, což je prostor obklopující znakující osobu od hlavy po pas a po stranách po roztažené lokty.[4]

Přestože se znakové jazyky vyvíjejí po boku mluvených jazyků, svoje protějšky nenapodobují. Například anglický znakový jazyk jen velice málo následuje posloupnost času, místa, podmětu, předmětu, slovesa a otázky. Místo toho je charakterizován dvojicí topic a comment, kde topic je téma, které je dáno na začátek věty a poté je okomentováno. Používá svojí vlastní syntaxi, která využívá jak manuální tak nemanuální charakteristiky, simultánní a posloupné vzorce a prostoré a stejně tak i lineární uspořádání. [5]

2.3.1. Lineárnost a simultánnost

Dalším rozdílem mezi jazyky je jejich linearita a simultaneita. Mluvený jazyk používá jednotky řazené výhradně lineárně za sebou. U jazyka znakového se ve velkém množství používá díky trojrozměrnému prostoru simultánní princip. Fonologické znakové jednotky, jako je tvar ruky, orientace dlaně a prstů a pohyby ruky se vzájemně překrývají a jsou znakované ve stejný okamžik. Kromě simultánnosti používá znakový jazyk také lineární řazení jednotek jako je u všech mluvených jazyků.[4] Principy lineárnosti a simultánnosti ve velké míře ovlivňují rychlost převodů myšlenkových obsahů. Znakování jednotlivých znaků je sice pomalejší než produkce jednotlivých slov, nicméně díky simultánnosti znakového jazyka je rychlost znakování celých výpovědí stejně dlouhá jako rychlost sdělení celých výpovědí v mluveném jazyce.[6]

2.4. Složky znaku

Stejně jako jsou slova tvořena slabikami a písmeny, jsou znaky tvořeny menšími fonologickými jednotkami.Tím se liší znaky od gest, které se z dalších částí neskládají.

Mezi tyto jednotky podle Stokoea patří umístění znaku v prostoru, tvar ruky a pohyb ruky v prostoru. V pozdější době byly jednotky rozšířeny o orientaci dlaně a prstů a vzájemnou polohu rukou.

Vedle těchto složek znaku manuálního charakteru jsou znaky rozšířeny o složky nemanuálního charakteru. Tyto nemanuální signály (NMS) mají nepostradatelnou roli pro tvoření syntaktických vztahů. Jsou znakovány simultánně spolu s manuálními komponenty. Mezi tyto nemanuální složky patří mimika obličeje, pohyb trupu a pohyb hlavy. Mimika je složka charakterizována pohyby mimických svalů obličeje. Tyto mimické pohyby nejsou nijak náhodné a i když se podle charakteru mluvčího od sebe odlišují, mohou být nositeli ustálených výrazů. Pohyb trupu vyjadřuje např. výběr mezi dvěma možnostmi nebo podmínku. Pohyby hlavy jsme pak schopni rozlišit pozitivní sdělení od záporného, kývání hlavy vyjadřuje oznamovací způsob.[4]

Důležitost NMS znázorňuje příklad věty z Amerického znakového jazyka na obrázku (obr. 2.1) Z pozorování gestikulace rukou lze vyvodit lexikální význam *"Ty studuješ."* Nicméně bez pozorování NMS a stylu znakování nejsme schopni rozpoznat pravý význam věty který zní: *"Studuješ pilně?"* Otázka ve větě je vyjádřena tělem sklánějícím se dopředu, kývnutím hlavy a pozdviženým obočím na konci znakující sekvence. Pro zdů-

Obrázek 2.1: Video zobrazující větu přeloženou do češtiny jako "Studuješ pilně?"[7]

raznění, že je aktivit dělána s velkou intenzitou, jsou rozevřené rty a výrazně zaťaté zuby. Kromě zohlednění NMS je navíc gestikulace rukou znázorňována opakovaně aby byla zvýrazněna průběhovost děje.[7]

Pro slova cizího původu, jmen, odborných termínů, pojmů a dalších slov, pro která ještě neexistují v ČZJ ustálené znaky, se využívá prstová abeceda. Prstová abeceda je lineární forma znakování, která využívá ustálených postavení dlaně a prstů jedné ruky k zobrazování jednotlivých písmen české abecedy. http://ruce.cz/clanky/3-prstova-abeceda

2.5. Charakteristiky znakového jazyka

Níže jsou popsány některé charakteristiky ČZJ. Vzhledem k tomu, že není možné popsat veškerou strukturu jazyka, se výběr zaměřuje na ty aspekty, které tvoří jistou výzvu v poli detekce znakového jazyka.

(a) Příslovce modifikující sloveso; pro výraz běžet rychle se nevyužívá dvou znaků, ale využije se znak pro výraz běžet, a ten se zrychlí.

(b) Nemanuální složky; výraz obličeje a postoj těla jsou klíčem v určování významu vět. Například pozice obočí může znamenat charakter otázky. Některé znaky se od sebe odlišují pouze tvarem rtů, protože sdílejí stejné manuální rysy.

(c) Umístění; zájmena jako on, ona nebo ono nemají svůj vlastní znak, místo toho je zástupci popsána a přidělena pozice ve znakovém prostoru.

(d) Klasifikátory; jsou to tvary ruky, které jsou použity ke znázornění tříd předmětů. Jsou použity, když dříve popsané věci spolu reagují. Například rozlišení mezi osobou honící psa a naopak.

(e) Směrová slovesa; odehrávají se mezi znakující osobou a referentem, směr pohybu naznačuje směr slovesa. Dobrým příkladem směrových sloves jsou slovesa dát a telefonovat. Směr slovesa implicitně vyjadřuje, které podstatné jméno je podmět a předmět.

(f) Poziční znaky; znak působí popisně na části těla, například modřina nebo tetování.

(g) Posun těla; představovaný kroucením ramen a upřeným pohledem, často používaný ke znázornění změny role vztahující se k dialogu.

(h) Ikoničnost; znak napodobuje věc kterou reprezentuje. Může být změněn, aby dával odpovídající smysl. Například znak pro vstávání z postele se může od sebe lišit mezi rychlím energickým vyskočením z postele a pomalým neochotným vstáváním.

(i) Prstová abeceda; tam kde není znak známý, ať už znakující osobou nebo adresátem, může být mluvené slovo pro daný znak vyhláskováno pomocí prstové abecedy.[5]

3. Detekce pomocí rukavic

Získávání dat je první krok pro detekci znakového jazyka (DZJ). Mnoho raných systémů DZJ používá datové rukavice a akcelerometry pro získání specifikací pohybu rukou. Měření (pozice x, y, z, orientace, rychlost atp) byla měřena přímo s použitím senzorů jako je Polhemův tracker a datová rukavice.[5]

Při porovnávání s metodami využívající video mají datové rukavice několik výhod i pár nevýhod. Jejich výhody jsou:

(a) Rukavicové systémy jsou cenově dostupnější než video systémy, obzvláště při současném vylepšení v technologii rukavic.

(b) Požadavky na výpočetní sílu pro zpracování videa v reálném čase jsou vysoké. Data extrahovaná z rukavice jsou stručná a přesná při porovnávání s informacemi z kamery

(c) Jistá data, jako jsou orientace ruky, pohyb dopředu a dozadu a pozice prstů, se velice těžce z vizuálních obrázků extrahují.

(d) Rukavice mohou být potenciálně užity beze změny prostředí, zatímco kamery potřebují být nastaveny do prostředí.

Na druhou stranu rukavice jsou pro uživatele přítěž. Rukavice snímají pouze pohyby rukou a neberou v potaz ostatní složky znakového jazyka jako je mimika obličeje. Je tu také pár dalších problémů, které potřebují vyřešit, jako je automatická kalibrace a vypořádání se s šumem.[8]

3.1. Hardware

Konstrukce rukavic se v různých provedeních mírně liší. Nicméně většina obsahuje prvky které se spolu schodují.

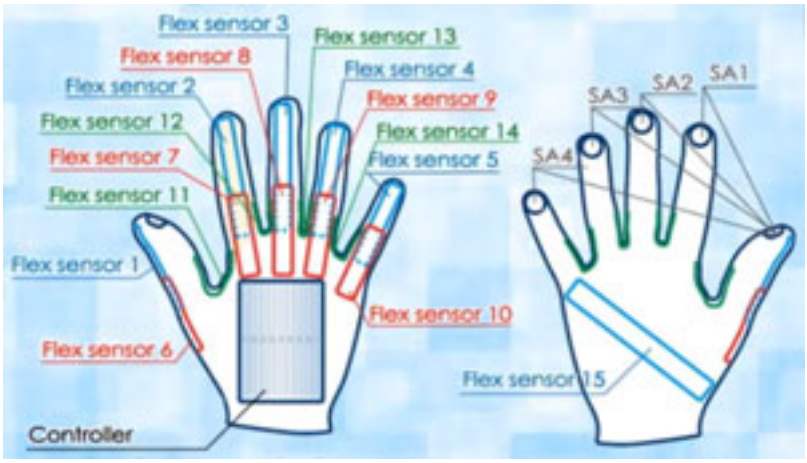
V experimentu od Mohammed Waleed Kadous [8] byla použita komerčně vyráběná rukavice Power Glove, která zaznamenává souřadnice x,y,z vzhledem k bodu, se kterým je rukavice synchronizována. Každá souřadnice může nabývat 255 různých hodnot. Dále zaznamenává natočení zápěstí s nárůstem 30 stupňů a ohyb čtyř prstů. Tato rukavice však nemá senzory na malíčku, což snižuje její funkčnost.

Další sériově vyráběna rukavice CyberGlove je opatřena 22 senzory, z nichž na každém prstě jsou tři. Dále je vybavena čtyřmi senzory pro měření odchýlení, jedním dlaňovým senzorem a senzory pro měření ohybu zápěstí. CyberGlove motion capture system byl využit v řadě reálných aplikací zahrnující virtuální realitu, biomechaniku a animaci.[9]

Ostatní rukavice jsou již prototypy a nejsou sériově vyráběné. Každá z nich disponuje, stejně jako rukavice vyráběné sériově, flex senzory, které jsou připevněné na prsty a snímají jejich ohyb. Dále rukavice disponují trojosým akcelerometrem a gyroskopem, které získávají data o pozici ruky v prostoru a o rotačním pohybu. Výrazný úspěch zaznamenal projekt ukrajinských studentů Enabletalk, který vyhrál první cenu v soutěži Microsoft Imagine Cup. V jejich rukavici je celkem 15 flex senzorů,kompas, akcelerometr a gyroskop. Data ze senzorů jsou zpracována microkontrolerem uchyceným na ruce, pak jsou přes bluetooth poslána do mobilního zařízení, které překládá do textu, když je rozpoznán

Obrázek 3.1: CyberGlove[9]

vzorec pohybu rukou a prstů.[10] Rozmístění senzorů na rukavici je zobrazen na obrázku obr.3.2



Obrázek 3.2: Rozmístění flex senzorů[10]

3.2. Software

3.2.1. Strojové učení

Strojové učení(SU), jak už napovídá název, je o schopnosti počítače se učit. Jedním z tradičních formalismů SU je kategorizace a klasifikace, kde je nám dán objekt, který sdílí podobné vlastnosti (nebo atributy) a my víme k jakému typu (nebo třídě) objektů patří. Naším cílem je najít způsob jak klasifikovat nový objekt neznámé třídy.

Pro tento případ bylo formulováno mnoho technik včetně instance-based learning, neuronové sítě, rule-learning systémy, systému stromového rozhodování, genetických algoritmů a induktivního logického programování. Každá odvozuje a vyjadřuje svoje klasifikační schéma jiným způsobem.

Kvalita rozpoznávání je zcela zřejmě spojena s atributy, které poskytujeme klasifikátoru.Většina objektů našeho zájmu má mnoho možných atributů, z nichž je jich pár vhodný pro klasifikaci. Selekce atributů je proto velice důležitá.[8]

3.2.2. Extrakce rysů

Jak již bylo zmíněno, extrakce rysů je pro úspěšný rozpoznávací proces kriticky důležitá. Tedy klíčovou částí v rozpoznávání je najití souboru atributů, které přesně popisují znak. Ve více případech se setkáváme s tím, že senzorové vstupy měly dostatečné hodnotové rozlišení. Extrakce rysů byla obejita a měření byla použita přímo jako funkce.[5]

V práci od Mohammed Waleed Kadou [8] byly za pomocí instance-based learning a systému stromového rozhodování testovány následující rysy. Přesnost každého rysu byla určena individuálně.

Vzdálenost, energie a čas

Vzdálenost, která se urazí při znázorňování znaku, může být jistě dobrým diskriminantem. Mimoto, přestože znaky mohou pokrývat podobné vzdálenosti, mohou být někdy gesta více energická, jako například dělání malých kruhů rukou. Byly užity jednoduché techniky pro aproximaci vzdálenosti a energie každého znaku. Některé znaky rovněž zabraly delší dobu na znázornění než jiné, což může být potenciálně dobrý atribut.

Ukázalo se ale, že se nejedná o správné řešení. Jeví se, že šum generovaný rukavicí dominuje v měření vzdálenosti a energie, a že čas potřebný pro znázornění znaku není dobrým diskriminantem. Přesnost dosažená těmito třemi atributy byla v průměru 8 procent.[8]

Hraniční kvádr

Hraniční kvádr znaku je kvádr v prostoru, do kterého se vejde celý vyjadřovaný znak. Hraniční kvádr je definován dvěma body: souřadnicemi dolního rohu kvádru u levé ruky a souřadnicemi horního rohu kvádru u pravé ruky.

Výsledky užívání hraničního boxu jsou dobré. Dosažená přesnost byla průměrně 30 procent. Hraniční kvádry zřejmě pracují dobře, protože nejsou citlivé na šum. Nicméně jsou náchylné na občasné chyby v měření, kdy jedna chyba může přerušit celý hraniční kvádr.[8]

Histogramy

Histogramy pracují na bázi segmentace určitého rozsahu hodnot do sub-regionů, ve kterých zůstávají po určitý čas. Například můžeme najít, že hodnota na ose x se pohybovala mezi 0 a 1 po dobu 60 procent času a mezi 1 a 2 po dobu 40 procent času.

Komplikovaným problémem histogramů pro SU je optimální počet dělení. Jestliže budeme mít příliš mnoho oddílů, šum bude interferovat s hodnotami a bude zodpovědný za příliš velkou citlivost. Když budeme mít příliš málo oddílů, znak nebude pro svoji detekci dostatečně charakterizován.

Histogramy byly kalkulovány pro následující zdroje informací:

Pozice x, y, z: dosažená přesnost 15 až 25 procent.

Rotace zápěstí a ohyb prstu: dosažená přesnost 30 až 40 procent.

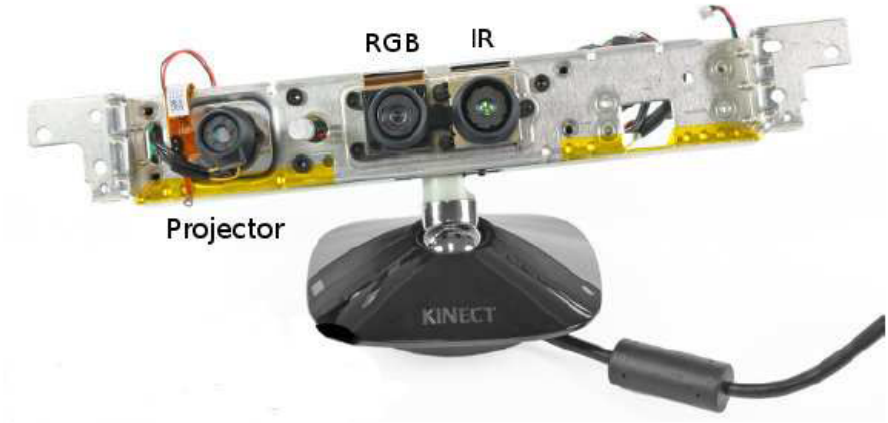
Vzdálenost a energie: dosažená přesnost 4 procenta[8].

3.2.3. Zhodnocení

Rukavice představují funkční hardware pro detekci znakového jazyka. V práci od Mohammed W. K. byla dosažena přesnost měření 80% za užití techniky instance-based learning.[8] V budoucnu je možný další vývoj rukavic a zpřesňování jejich měření, nicméně jejich rozšíření do běžného života neslyšících lidí se nepředpokládá vzhledem k tomu, že jsou při znakování nepraktické.

4. Kinect

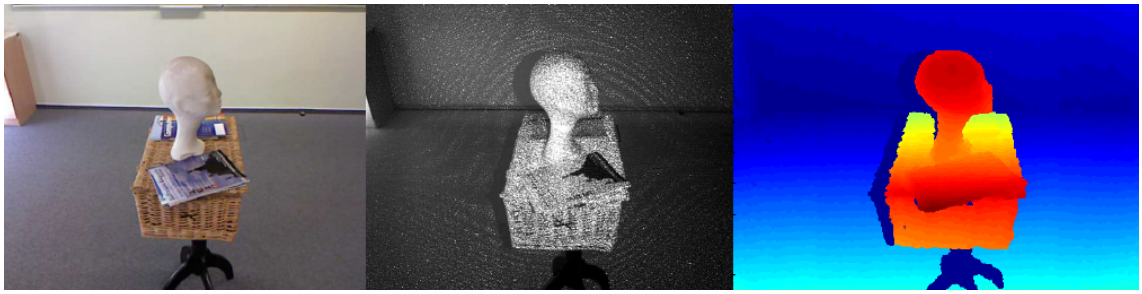
Kinect se v poslední době stal důležitým 3D senzorem. Dostalo se mu velké pozornosti díky rychlému rozvinutému rozpoznávání lidských póz a také 3D měření. Spolehlivost a rychlost měření dělají z Kinectu primární 3D měřící zařízení v robotice, scénové rekonstrukci a objektovém rozpoznávání.[11] Sériová výroba Kinectu nicméně před několika lety skončila.



Obrázek 4.1: Kinect[11]

4.1. Hardware

Kinect je kompozitní zařízení skládající se z infračerveného (IR) laserového projektoru, IR kamery a barevné (RGB) kamery o rozlišení 1280×1024. IR kamera a projektor jsou užity jako stereo pár pro generování triangulačních bodů ve 3D prostoru. RGB kamera může být pak užita pro texturování 3D bodů nebo k rozpoznávání obrazového obsahu. Jako měřící zařízení dodává Kinect tři výstupy: IR obraz, RGB obraz, a hloubkový obraz.[11]



Obrázek 4.2: Obraz z RGB kamery, IR kamery a hloubkový obraz[11]

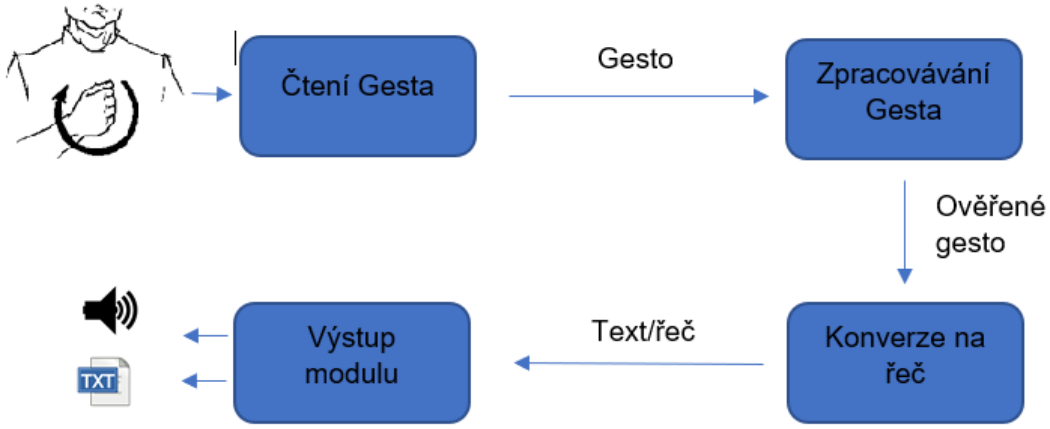
4.2. Software

Tato část obsahuje implementační techniky užité v modulu, který převádí znaky do slov. To se provádí pomocí Microsoft Kinect V2 Continuous Gesture Builder, který je odpovědný za rozpoznávání gest a učení. Implementační detaily modulu jsou uvedeny níže.

4.2.1. Implementační přístup

Kinect v2 SDK (*Software Development Kit*)je opatřen nástrojem nazvaným gesture builder, který je konkrétně navržený pro jednoduché rozpoznávání gest. Tento nástroj se učí na již uložených gestech a detekuje znázorňovaný znak, když uživatel pohybuje rukou. Nástroj gesture builder využívá daty řízené algoritmy strojového učení pro trénování gest.[12]

Pro implementaci tohoto modulu byly použity nástroje a knihovny poskytnuté od firmy Microsoft for Kinect development, které zahrnují Kinect studio pro nahrávání gest ve specifickém formátu, Visual Gesture Builder (VGB) pro učení gest, pro něž může být generována databáze gest. Implementační přístup modulu, který převádí znaky do slov, je popsán na obrázku 4.3.



Obrázek 4.3: Pracovní postup modulu převádějícího znaky do řeči[12]

Systém dostane na vstupu gesto snímek po snímku za použití Kinectu a porovná každé gesto s předuloženými gesty v databázi. Pro každé spárované gesto je přiřazeno klíčové slovo. Poté je pomocí těchto klíčových slov sestavena věta a nakonec je věta konvertována na řeč za použití knihoven .Net. Poté je věta přeříkána počítačem ve formě přirozeného jazyka, kterému každý dobře rozumí.[12]

4.2.2. Užité algoritmy

Technologie rozpoznávání gest je ve VGB rozdělena do dvou kategorií Adaboost a RFRProgress. Adaboost je spouštěč, který dává pravdivou booleanovou hodnotu, když osoba předvádí určité gesto. Využívá algoritmy strojového učení Adaptive Boost. RFRProgress na druhou stranu produkuje kontinuální výsledky dávající analogová data, zatímco uživatel předvádí gesto. Systém detekuje, kolik částí gesta je již hotovo a jak velký je podíl shody konkrétního snímku gesta. Využívá algoritmů strojového učení Random Forest Regression.

Pro správné rozpoznávání gesta byly v systému použity kombinace obou metod. Adaboost umožňuje systému detekovat začátek a konec klasifikovaného gesta. Vrací hodnotu true/false, zatímco je předváděno gesto. Když Adaboost vrátí hodnotu true, potom systém využije RFRProgress k detekování správného gesta čtením znázorňovaného gesta. V tomto případě je postup puštěn, je-li detekována shoda (Adaboost vrací hodnotu true).

Takže Adaboost určuje kontext předváděného gesta a RFRProgress detekuje gesto kontinuálním mapováním pohybu uživatele. Jestliže je podíl shody dostačující pro určité gesto kombinací metod Adaboost a RFRProgress, je výsledek považován za pozitivní.

Výsledná přesnost modulu s použitím výše popsaných algoritmů je 84%. [12]

5. Detekce pomocí obrazového záznamu

Od doby, kdy každé mobilní zařízení má schopnost nahrávat videa, je detekce znakové řeči z videového záznamu uživatelsky nejpřístupnějším způsobem detekce. Na druhou stranu tento způsob sebou nese řadu negativních faktorů ovlivňujících rozpoznávání znakového jazyka v reálném čase.

(a) Proces získávání obrazových záznamů klade mnoho nároků na okolní prostředí, např. umístění videokamery, citlivost na osvětlení, stav pozadí a počet použitých kamer.[13]

(b) Při znakování se ruce často překrývají mezi sebou i obličejem. Prsty nebo dokonce i celá ruka tak nemusí být zachycena.

(c) Hranice znaků musí být automaticky rozpoznány. Začátek a konec znaku musí být ze zaznamenaných dat videa automaticky zjištěn.

(d) Znak je ovlivněn předchozím i následujícím znakem (spolupůsobení).

(e) Umístění osoby před kamerou se může lišit. Výsledkem je nežádoucí časová a prostorová změna souřadnic osy. Musí být brán v úvahu pohyb osoby jako např. posun v jednom směru nebo rotace kolem osy těla.

(f) Každý znak se v čase a prostoru mění. Rychlost znakové řeči se výrazně liší. I pokud jedna osoba provede stejný znak dvakrát, dojde k malým změnám v rychlosti a poloze rukou.

(g) Projekce 3D scény na 2D rovině vede ke ztrátě informací o hloubce. Obnovení 3D trajektorie ruky v prostoru není vždy možná.

(h) Zpracování velkého množství obrazových dat je časově náročné, rozpoznání v reálném čase je proto obtížné.

(i) Vyšší rozlišení způsobuje značné zpoždění při získávání obrazových záznamů a delší čas zpracování.

(j) Při zpracování v reálném čase musí být překladač dostatečně rychlý v pořizování snímků osoby, zpracování obrazů a zobrazení překladu znaků na obrazovce počítače.

Zpracování videa v reálném čase je poměrně náročné. Při zpracovávání mluvíme o dvou snímkovacích frekvencích. Jedna se týká snímkovací frekvence kamery a druhá rychlosti zpracovávání obrazu výpočetními algoritmy.[14] Pro zpracovávání videa v reálném čase musí být snímkovací frekvence kamery nižší než je frekvence zpracovávání obrazu. Celková snímkovací frekvence je pak ta menší ze dvou frekvencí.

Při klasifikaci se setkáváme s dvěma typy znaků, a to se statickými a dynamickými znaky. Statické znaky zahrnují časově nezávislé orientace prstů, zatímco dynamické znaky zahrnují časově proměnné orientace rukou a pozice hlavy.[15] Pro klasifikaci to znamená, že u statických znaků stačí klasifikovat jediný obraz zatímco u dynamických je nutné klasifikovat celou sekvenci obrazů, které zachycují daný znak.

5.1. Segmentace videa

Segmentací videa rozumíme zjednodušení jednotlivých obrazů do stavu, kdy budou jednodušeji analyzovatelné pro svou další klasifikaci. Pro analýzu znakového jazyka jsou nejdůležitějšími částmi těla ruce. Segmentace rukou může být provedena na základě detekce kůže. Zachycený obraz ruky je z formátu RGB konvertován do formátu YCrCb kvůli redukci změny osvětlení v prostředí.[16] Konverze je dosažena následujícím vztahem 5.1.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0,29900 & 0,587000 & 0,114000 \\ -0,168736 & -0,331264 & 0,500000 \\ 0,500000 & -0,418688 & -0,081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5.1}$$

Detekce barvy kůže je dosažena na základě hodnot dvou chromačních komponentů Cr a Cb a zanedbáním komponentu jasu Y.[16] Pro detekci kůže platí vztah 5.2

$$w(x,y) = \begin{cases} 255, if Cr1 < Cr < Cr2 and Cb1 < Cb < Cb2 \\ 0, others \end{cases} \tag{5.2}$$

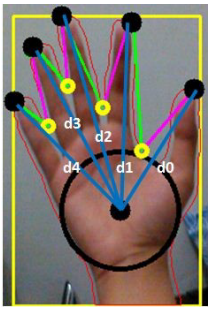
Každá hodnota pixelu z obrazu, která zapadá do intervalů Cr a Cb, bude považována za kůži a bude mu přiřazena hodnota 255 (bílá), zatímco hodnota mimo tento rozsah bude považována za pozadí a bude mu přiřazena hodnota 0 (černá).[16] Výsledkem tohoto procesu je segmentovaná oblast ruky v bílé barvě a zbytek je považován za pozadí v černé barvě, jak je zobrazeno na obrázku 5.1.

Obrázek 5.1: Segmentovaný obraz s detekcí kůže[16]

Dalším krokem je najít obrysy ze segmentovaného obrazu vyplývající z detekce kůže.[16] Pro nalezení obrysů ze segmentovaného obrazu je použit algoritmus convex hull, který slouží k získání hranice ruky. Pro detekci prstů na ruce je užit algoritmus convexity defect.

5.2. Extrakce rysů

Tento proces získá všechny potřebné rysy z každého snímku gestikulované ruky. Ty jsou posléze klasifikovány a rozpoznávány pomocí neuronové sítě. Při detekci prstové abecedy, tedy v případě, kdy je v obrazu zaznamenána jen jedna ruka, jsou rysy ruky zastoupeny vzdáleností mezi konci prstů a středem dlaně a úhlem mezi sousedícími prsty při natažené pozici prstů. [16] Střed dlaně je nalezen pomocí středu hraničního boxu, který obklopuje ruku v obrazu. Dalším krokem je nalezení konečků prstů pomocí convexity defect algoritmu. Výsledkem tohoto procesu je vykreslení čar od konečků prstů do středu dlaně a nalezení úhlů sousedících prstů, jak je znázorněno na obrázku 5.2.



Obrázek 5.2: Výsledek extrakce rysů obrazu ruky[16]

Dalším krokem je oindexování každé čáry, která spojuje konečky prstů se středem dlaně, a změření délky každé čáry pomocí 5.3, kde x1 a y1 jsou souřadnice středu dlaně a x2, y2 jsou souřadnice konečku prstu.[16]

$$\sqrt[2]{(x1 - x2)^2 + (y1 - y2)^2} \tag{5.3}$$

5.3. Klasifikace

Klasifikace videa nebo jiného obrazového záznamu je náročný problém, jelikož sekvence videa obsahuje jak časově tak i prostorově proměnná data.[17] Prostorová data jsou získána ze snímků videa, zatímco časově proměnná data jsou sesbírána ze snímků v závislosti na čase. Při trénování modelu jsou využity dva typy neuronových sítí. Při trénování modelu na prostorová data se využívá konvoluční neuronová síť (CNN) a pro časově proměnná data se využívá rekurentní neuronová síť (RNN). Vstupem do těchto neuronových sítí jsou nijak nesegmentované digitální obrázky typu RGB. Pro klasifikaci segmentovaných obrázků typu 5.2 stačí plně propojená neuronová síť. Jejím vstupem jsou jednoduché číselné hodnoty jednotlivých úhlů a vzdáleností a ne celý obraz.

5.3.1. Backpropagtion

Backpropagation (BP), česky zpětné šíření, je algoritmus, který modifikuje váhové spojení mezi jednotlivými vrstvami v opačném směru gradientu chybové funkce, *error function*. Tímto způsobem se získá nový váhový vektor. Tato modifikace váhového vektoru se opakuje do doby, než chyba dosáhne nastaveného limitu. Váhy jsou aktualizovány podle následujícího vzorce:

$$w^{m+1} = w^m + \alpha \cdot (-\nabla^m) \tag{5.4}$$

kde α je učící krok, který nastavuje jak rychle se má síť učit a ∇ je gradient chybové funkce.

V algoritmu BP je použita střední kvadratická chyba, která je vypočítaná z požadovaného výstupu d^m jako:

$$(e^{m+1})^2 = (d^m - w^m \cdot x^m)^2 \tag{5.5}$$

Gradient je získán ze střední kvadratické chyby:

$$\nabla^m = -2 \cdot e^m \phi(v^m) \cdot x^m \tag{5.6}$$

Z rovnice 5.4 pak dostáváme výraz:

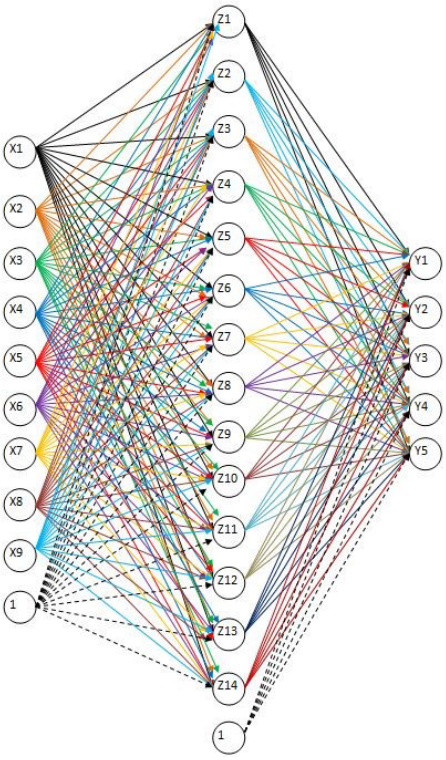
$$w^{m+1} = w^m + 2 \cdot \alpha \cdot \phi(v^m) \cdot x^m \tag{5.7}$$

Tento proces je aplikován na všechny neurony každé vrstvy sítě.

5.3.2. Plně propojená neuronová síť

Plně propojená neuronová síť je jednoduchý klasifikátor, který za pomoci zpětného šíření, anglicky backpropagation, dokáže efektivně klasifikovat jednotlivé znaky. Samotný proces BP je časově náročný, ale po jejím provedení je již klasifikace rychlá. Je vhodná pro klasifikaci méně náročných znaků jako je prstová abeceda.[16]

V experimentu byla klasifikována indonéská prstová abeceda o dvaceti šesti písmenech, z nichž ale dvě písmena jsou dynamické povahy, takže jsou pro tento typ klasifikace nevhodná. Zbylých 24 písmen se znakuje staticky. V experimentu je neuronová síť tvořena třemi vrstvami. Vrstva na vstupu se skládá z devíti neuronů, z nichž pět tvoří vzdálenosti konců prstů od středu dlaně a čtyři tvoří úhly mezi prsty. Prostřední vrstvou je skrytá vrstva, kterou tvoří 14 neuronů, jejichž hodnoty jsou na začátku náhodné. Po BP se jejich hodnoty nastaví do funkční hladiny. Výstupní vrstva je tvořena pěti neurony, které jsou vyjádřeny v binární formě a číselně vytváří 32 čísel. Pro pokrytí abecedy, kde každé číslo bude zastupovat jedno písmeno, jich jev tomto případě potřeba pouze 24. Každý neuron je propojen s každým neuronem sousedící vrstvy a každé toto spojení má svou číselnou váhu, která se během BP upravuje.[16] Struktura neuronové sítě je zobrazena na obrázku 5.3.



Obrázek 5.3: Plně propojená neuronová síť[16]

Aby síť fungovala, musí být nejdříve trénována se vstupními a výstupními daty. Teprve potom může být použita ke klasifikaci vstupních dat, které reprezentují písmeno z

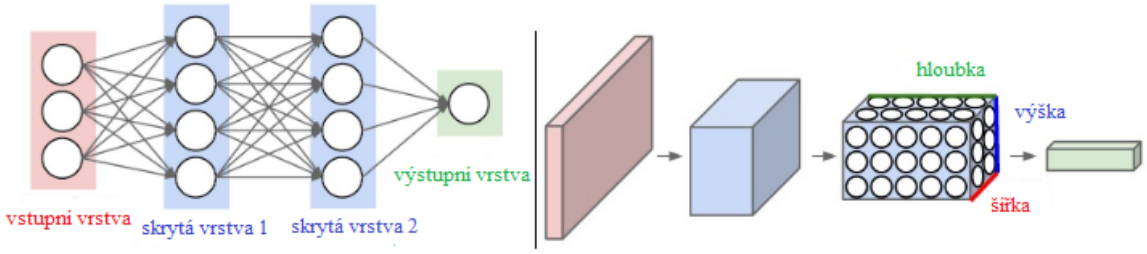
prstové abecedy. Míra učení je nastavena na 0,1 , 0,2 a 0,3. Kvůli výpočtové náročnosti byl trénovací proces pro získání váhových hodnot proveden na počítači. Po fázi trénování byly získané hodnoty neuronové sítě uloženy a poté mohly být použity v praxi nahráním přes aplikaci na chytrý telefon a testován v reálném čase.

5.3.3. Konvoluční neuronová síť (CNN)

Konvoluční neuronové sítě jsou hluboké umělé neuronové sítě, které se užívají primárně na klasifikaci obrázků.[18] Konvoluční neuronové sítě jsou výborné v zachycování lokálních prostorových vzorů v datech.[17] Jsou účinné v hledání vzorů, které jsou poté využity ke klasifikaci obrazů. CNN výslovně předpokládají, že vstupem do sítě bude obrázek. CNN jsou díky poolingovým vrstvám necitlivé na rotaci obrazu, takže obraz a jeho natočená verze obrazu budou klasifikovány stejně. Příkladem hluboké CNN, která využívá širokých výhod extrahování prostorových rysů z obrázku, je model Inception-v3[19] s knihovnou Tensorflow[20]. Inception-v3 je rozsáhlý model na klasifikaci obrazu s miliony parametrů pro klasifikaci.

Architektura CNN

Konvoluční neuronové sítě jsou podobné běžným neuronovým sítím. Jsou složeny z neuronů, které mají učitelné váhy a biasy. Každý neuron dostane jistý vstup, provede skalární součin a volitelně jej následuje nelinearitou. Na rozdíl od běžných neuronových sítí mají vrstvy CNN neurony uspořádány do třech dimenzí: šířka výška a hloubka. Například obrázky z databáze CIFAR-10, které mají rozměry 32x32x3 (šířka výška, hloubka), jsou klasifikovány tak, že neurony ve vrstvě jsou spojeny pouze s malou oblastí neuronů ve vrstvě před ní, místo toho, aby byly propojeny se všemi neurony jako tomu je u plně propojených sítí. Ve výsledku má výstupní vrstva pro CIFAR-10 dimenze 1x1x10, protože na konci struktury CNN se obrázek zredukoval z plného obrázku na jediný vektor. Hodnoty tohoto vektoru pak odpovídají deseti třídám, které se tímto systémem klasifikují.[21] Vizualizace je zobrazena na obrázku 5.4



Obrázek 5.4: Vpravo plně propojená neuronová síť, vlevo schéma konvoluční neuronové sítě[21]

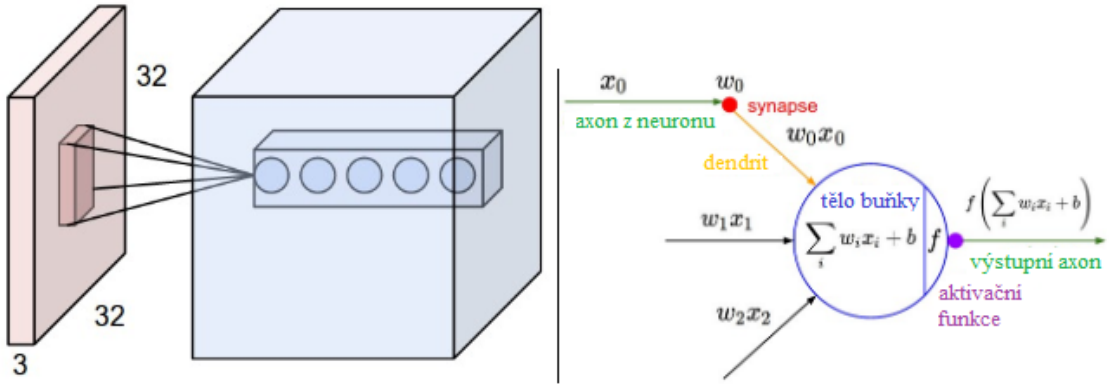
Jak bylo popsáno výše, jednoduchá CNN je sekvencí vrstev a každá vrstva CNN transformuje objem aktivací na jiné přes diferencovatelnou funkci. Na stavbu CNN se využívají tři hlavní typy vrstev: konvoluční vrstva, poolingová vrstva a plně propojená vrstva. Tyto vrstvy se skládají za sebou, aby se zformovala plná konvoluční síť. [21]Příkladem může být architektura jednoduché CNN pro klasifikaci z databáze CIFAR-10:

- Vstup [32x32x3] obsahuje syrové hodnoty pixelů z obrázku, v tomto případě šířky 32, výšky 32 a hloubky 3 pro každou barvu RGB.
- Konvoluční vrstva počítá výstup neuronů, které jsou spojeny s lokálními oblastmi na vstupu. Každý spočítá skalární součin mezi svými váhami a malou oblastí, se kterou jsou spojeny. Výsledkem tohoto může být objem například [32x32x12], pokud bude použito 12 filtrů.
- RELU vrstva aplikuje na každý element aktivační funkci ve tvaru $max(0, x)$. Tento postup nechává rozměry objemu nezměněný ([32x32x12]).
- Poolingová vrstva způsobuje podvzorkovávací operaci přes prostorové rozměry (šířka výška), jejímž výsledkem je objem [16x16x12].
- Plně propojená vrstva vypočítává výsledky tříd výsledným objemem [1x1x10], kde každé z deseti čísel koresponduje s výsledkem třídy. Jak název napovídá, každý neuron je propojen se všemi neurony v předešlé vrstvě.[21]

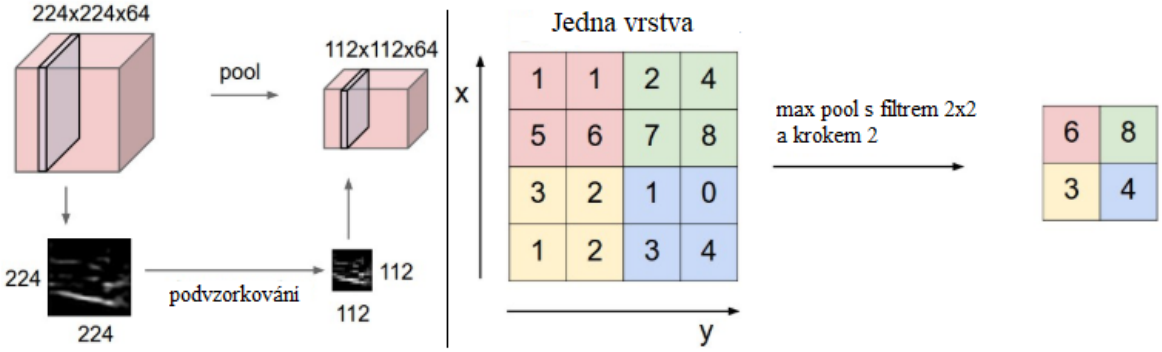
Tímto způsobem CNN transformuje originální obrázek vrstvu po vrstvě z originálních hodnot pixelů do finální podoby výsledků jednotlivých tříd. Konvoluční a plně propojené vrstvy obsahují parametry (váhy a biasy), které vstupují do aktivační funkce spolu se vstupními hodnotami z předešlé vrstvy. Na druhou stranu RELU a poolingové vrstvy implementují pouze fixní funkci. Parametry z konvoluční a plně propojené sítě jsou trénovány s gradientním sestupem tak, aby výsledky tříd ve výstupní vrstvě, kterou CNN vypočítá, korespondovaly s labely v trénovacím setu pro každý obrázek.[21]

Konvoluční vrstva Parametry konvoluční vrstvy se skládají ze sady filtrů, které mají možnost učení. Každý filtr má malé prostorové rozměry (šířka a výška), které prostupují do celé hloubky vstupního obsahu. Typický filtr konvoluční vrstvy má rozměry 5x5x3, kde rozměr 5 zastupuje výšku a šířku pixelů a 3 zastupuje hloubku barevných kanálů RGB. Filtry poté přejíždějí po šířce a výšce vstupního obsahu a vypočítávají skalární součin mezi parametry na aktuální pozici. Výsledkem takto přefiltrovaného obrazu je dvourozměrná aktivační mapa. Parametry filtru se učí, když zaznamenají nějaký druh vizuálního rysu, jako je hrana nějaké orientace nebo skvrnu nějaké barvy. Při použití více filtrů, každý filtr vyprodukuje jednu aktivační mapu. Tyto mapy se poté seřadí za sebou a vytvoří výstupní obsah. Při zpracovávání vícerozměrných vstupů jako jsou obrázky, je nepraktické spojovat každý neuron se všemi neurony v předchozí vrstvě. Místo toho je každý neuron spojen pouze s lokální oblastí vstupního obsahu. Popis konvoluční vrstvy je zobrazen na obrázku 5.5

Poolingová vrstva Je běžné periodicky přidávat poolingové vrstvy mezi konvoluční vrstvy. Její funkcí je progresivně snížit prostorovou velikost vstupního obsahu, aby se snížil počet parametrů a výpočetní nároky sítě. Poolingová vrstva funguje nezávisle na každé hloubkové vrstvě vstupu a změní její velikost prostorově za užití MAX operátoru. Nejběžnější formou je poolingová vrstva s filtrem o velikosti 2x2 aplikovaný s krokem 2, čímž se redukuje velikost každé vrstvy na vstupu dvakrát, což způsobí vyřazení 75 procent aktivací. Každá MAX operace v tomto případě vybere nejvyšší hodnotu ze čtyř čísel. Hloubka obsahu se mezi vstupem a výstupem nemění. Vizualizace poolingových vrstev je zobrazena na obrázku 5.6



Obrázek 5.5: Vlevo schéma konvolční neuronové sítě s filterm. Každý neuron je spojen pouze s malopu oblastí vstupního obsahu.Vpravo výpočetní schéma neuronu [21]



Obrázek 5.6: Vlevo, vstup o rozměrech [224x224x64] je poolingovým procesem zredukován na rozměr [112x112x64]. Hloubka je zachována. Vpravo, nejběžnější podvzorkovávací operace MAX. [21]

5.3.4. Rekurentní neuronová síť (RNN)

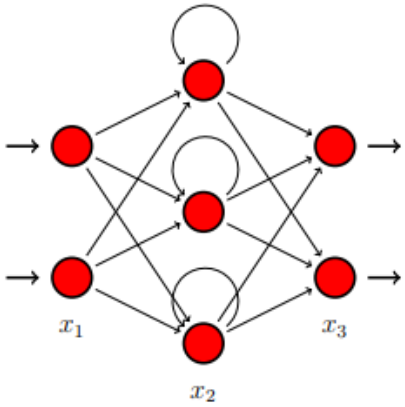
V samotné sekvenci obrázků existuje informace a toho využívají rekurentní neuronové sítě (RNN) pro rozpoznávání znaků.[17] Výstup RNN závisí na kombinaci současného vstupu a předchozího výstupu, takže funguje ve smyčce. Nevýhodou RNN je, že se v praxi nedokáže naučit dluhodobé souvislosti. [22] Aby se tomuto předešlo, může model obsahovat paměť *Long Short-Term Memory* (LSTM), což je variace RNN s prvky LSTM. LSTM se může naučit přemostit interval až o tisíc kroků i v případě sekvence vstupu s velkým šumem.[17] Schéma rekurentní neuronové sítě je na obrázku 5.7

Aktivace rekurentní vrstvy je formulována jako:

$$h_t = \sigma(Wx + Uh_{t-1} + b) \tag{5.8}$$

kde x je aktivace předchozí vrstvy v síti a W je přidružená matice váhových spojení. Aktivace z předešlého kroku rekurentní sítě h_{t-1} je vynásobena s váhovou maticí U. σ je aktivační funkcí a b je biasový vektor příslušné vrstvy.[23]

RNN jsou zvláště citlivé na mizení gradientu, kdy neurální aktivace po více krocích saturují a velikost gradientu skončí na hodnotách blízké nule. Následující rozšíření RNN popsané níže tento problém potlačují.[23]



Obrázek 5.7: Rekurentní neuronová síť. Neurony v druhé vrstvě v sobě mají zpětnou smyčku, takže jejich aktivace závisí na předchozí aktivaci v čase. [23]

LSTM

Sepp Hochreiter a Jürgen Schmidhuber vyvinuli rozšíření pro rekurentní neuronové sítě nazvané Long Short-Term Memory (LSTM)[24]. Potlačuje problém mizejícího gradientu tím, že se učí, kdy modifikovat skrytou vrstvu za použití takzvaných bran *(gates)*. Existují tři typy bran: vstupní brána *(input gate)*, výstupní brána *(output gate)* a zapomínací brána *(forget gate)*. Každá z nich se chová jako samostatná jednovrstvá neuronová síť. Výstup každé propusti je omezen pouze na [0, 1] za užití aktivace sigmoid. Všechny propusti sledují jak předchozí skrytý stav, tak současný vstup do vrstvy LSTM. Výstupem zapomínací propusti je vektor, který se vynásobí s hodnotami skrytého stavu, kde výstup hodnoty nula způsobí vyřazení skrytého stavu a výstup hodnoty jedna ho zachová. Vstupní propust kontroluje, kdy využít informaci přítomnou na vstupu a výstupní propust kontroluje, jaký má vliv skrytý stav na výstup sítě.[23]

5.3.5. Metodologie

Trénování modelu na časová a prostorová data může být provedeno různými přístupy.[17] Příkladem jsou dva přístupy, oba dva se liší způsobem, jakým se dávají informace na vstupu do RNN na trénování časových vlastností.

Přístup predikce

V tomto přístupu byla prostorová data pro každý snímek extrahována za užití CNN a časová data za užití RNN. Každé video bylo poté reprezentováno sekvencí předpovědí provedených CNN pro každý individuální snímek. Toto bylo dáno jako vstup do RNN. Pro každé video odpovídající jednotlivým gestům byly extrahovány snímky a části pozadí, které nebyly ruce, byly odstraněny, aby se získal černobílý obraz rukou.[17]

Snímky ze sady na trénování byly dány do modelu CNN na trénování prostorových rysů. Získaný model byl poté použit, aby uchovával a dělal predikce pro snímky trénovacích a testovacích dat. Predikce korespondující se snímky z trénovacích dat byly poté dány do modelu RNN s LSTM pro trénování vlastností závislých na čase. [17]

Přístup poolingových vrstev

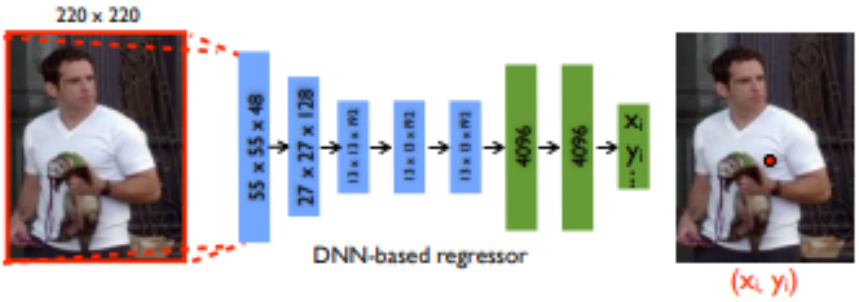
V tomto přístupu byla CNN užita pro trénování modelu na prostorové rysy. Výstup z poolingové vrstvy byl použit jako vstup do RNN dříve, než byla provedena predikce v CNN. Výstupem poolingové vrstvy je 2048 rozměrný vektor, který reprezentuje zkonvoluvané hodnoty z obrázku, ale ne predikci třídy, do které obrázek patří. Následující kroky v tomto přístupu se shodují s předchozím přístupem. Oba přístupy se pouze odlišují na vstupu který je dán do RNN.[17]

5.3.6. Detekce postoje těla a pozice rukou a paží

V této metodě se pomocí hluboké neuronové sítě (DNN) detekují pozice těla, rukou a paže, které jsou zobrazené pomocí úseček zalomených v kloubních spojeních těla. Tyto získané vlastnosti mohou být zjednodušením pro další klasifikaci neuronovými sítěmi. [25]

Problém odhadu postoje těla je definován jako problém lokalizace lidských kloubů. Tento problém je řešen pomocí sedmivrstvé konvoluční neuronové sítě, která má na vstupu obrázek v plném rozlišení. Tento postup má dvě výhody. Zaprvé, neuronová síť je schopna zachytit všechny souvislosti každého tělního kloubu - každý kloubový regresor používá plný obraz jako signál. Zadruhé, tento přístup je podstatně jednodušší na formulaci než metody založené na grafických modelech. [25]

Pro vyjádření pózy zakódujeme pozici každého tělního kloubu do vektoru vyjadřujícího pózu definovaného jako $y = (..., y_i^T, ...)^T$, kde y_i obsahuje x-ové a y-ové souřadnice í-tého kloubu. První vrstva neuronové sítě bere jako vstup obraz předdefinované velikosti a její velikost odpovídá počtu pixelů násobená třemi barevnými kanály. Z poslední vrstvy vystupují cílové hodnoty z regrese. V našem případě jsou to dvourozměrné souřadnice kloubů. [25]



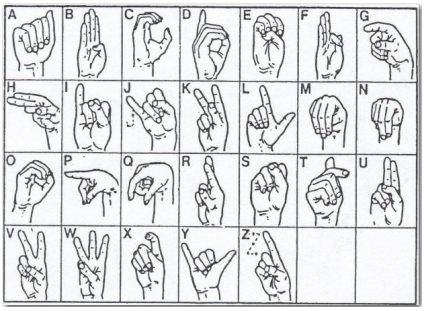
Obrázek 5.8: Schéma hloubkové konvoluční sítě[25]

Základní architektura je založena na práci vypracovanou Alexem Krizhevským et al. [26] na klasifikaci obrázků, jelikož má výjimečně přesné výsledky v oblasti lokalizace objektů. Síť se skládá ze sedmi vrstev. Jedné konvoluční vrstvy C, jedné vrstvy s lokální odezvou normalizace LRN, poolingové vrstvy P a plně propojené vrstvy F. Pouze C a F vrstva obsahuje učící parametry, zatímco zbytek parametrů je volný. Obě C a F vrstvy se skládají z lineárních transformací následující nelineární. Rozměry C vrstvy jsou definovány jako délka šířka a hloubka, kde první dvě dimenze mají prostorový význam, zatímco hloubka koresponduje s počtem filtrů. [25] Vizualizace pozice těla a rukou je zobrazena na obrázku 5.9.

Obrázek 5.9: Vizualizace pozice těla a rukou[25]

5.4. Detekce znaků prstové abecedy

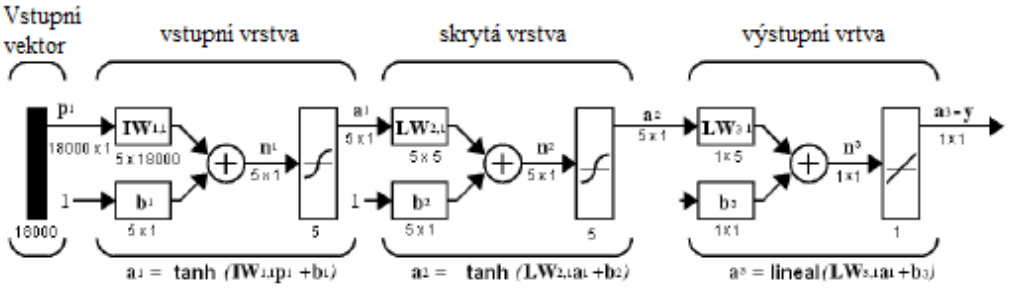
Detekce znaků prstové abecedy představuje oproti detekci ostatních znaků méně náročný problém vzhledem k tomu, že soubor písmen obecné latinky zastupuje pouze 26 znaků. V češtině se pak používá 42 znaků. Většina znaků je statické povahy, což je další zjednodušení, jelikož k jejich detekci postačí konvoluční nebo plně propojená neuronová síť. Česká znaková abeceda je zobrazena na obrázku 5.10. Níže jsou popsané různé postupy, které prstovou abecedu detekují.



Obrázek 5.10: Znaky české prstové abecedy[27]

5.4.1. Postup 1

Návrhem práce od Lorena P. Vargas je hardwarová implementace neuronové sítě za užití FPGA (*Field Programmable Gate Arrays*), která je použita na rozeznávání vzorců znakového jazyka. V návrhu je použita vícevrstvá neuronová síť s algoritmem zpětného šíření *Backpropagation*. Struktura sítě je tvořena třemi vrstvami, vstupní vrstvou, skrytou vrstvou a výstupní vrstvou.[28] Základní schéma sítě je zobrazeno na obrázku 5.11.



Obrázek 5.11: Použití vícevrstvé neuronové sítě[28]

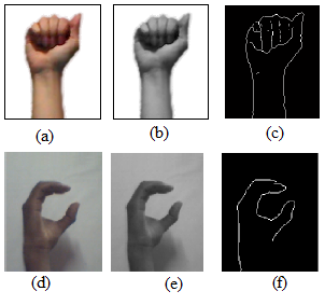
Pro neurony vstupní a skryté vrstvy byla použita aktivační funkce hyperbolický tangens tanh a pro neurony výstupní vrstvy byla použita lineární aktivační funkce. V trénovacím procesu prvního stupně byl použit algoritmus back propagation.

Výsledky

Po fázi učení je systém připraven rozpoznávat jednotlivé znaky. Pro vyhodnocení provedení implementovaného algoritmu je vyvinuto uživatelské prostředí v Matlabu, které dovoluje nahrát digitální obrazy, které jsou analyzovány a zaslány po sériích do FPGA.

K analyzování jsou použity obrazy o fixním rozměru 120x150 pixelů v černobílé barvě, kde každý pixel nabývá hodnoty 0 nebo 255.

Prvním procesem po uložení obrázků do paměti je jejich binarizace na černou a bílou a detekce hran.[28] Obrázek 5.12 zobrazuje výsledky po užití detekce hran.



Obrázek 5.12: a) ideální obraz, b) černobílý obraz, c) hrany ideálního obrazu, d) obraz s menším kontrastem a menším osvětlením, e) černobílý obraz s menším kontrastem, f) hrany obrazu s menším kontrastem [28]

Vzhledem k faktu, že vstupem do neuronové sítě musí být vektor, je každý testovací obrázek transformován tak, že je řádek po řádku rozložen za sebou, aby se zformoval ve vstupní vektor.

Protože vstupní obraz má rozměry 120 x 150, má vstupní vektor 18000 prvků, takže každý neuron vstupní vrstvy musí mít 18000 vah.[28]

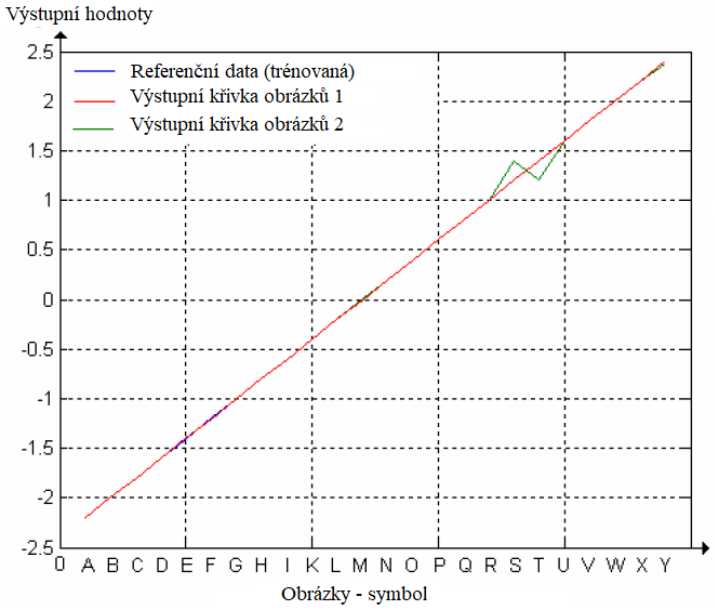
Graf na obrázku 5.13 zobrazuje výsledky za užití sítě se stejnou konfigurací jako na obrázku 5.11.

Ke klasifikaci byly použity dva typy obrázků a to s ideálním osvětlením a vysokým kontrastem a obrázky se špatným osvětlením a malým kontrastem. Každý symbol má v trénovací fázi přiřazenou výstupní hodnotu. Tyto hodnoty se od sebe liší po kroku velkém 0,2. Z obrázku 5.13 je vidět, že referenční data z trénovací fáze mají úzký vztah s daty získanými. Celková průměrná přesnost systému je 94% s rychlostí rozeznání znaku 60 ms. Graf zobrazuje, že došlo k chybě při klasifikaci písmen S a T. [28]

V této práci byl vyvinut systém pro rozeznávání obrázků se znaky prstové abecedy, ale není limitován pouze na tuto aplikaci. Návrh může být použit i na jiné typy znaků. V budoucnu je plánováno práci rozšířit o rozpoznávání dynamických znaků a znaků zachycených z různých úhlů pohledu.[28]

5.4.2. Postup 2

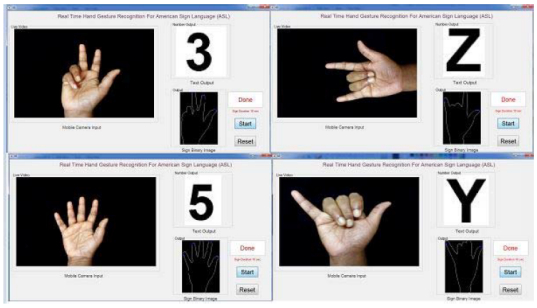
V této práci Md. Mohiminul Islam et al. [29] prezentuje systém pro detekci prstové abecedy z amerického znakového jazyka (ASL). Zpracovávané obrázky rukou mají černé pozadí a jsou pořízené z mobilní videokamery pro extrakci rysů. Ve fázi zpracovávání systém extrahuje pět typů rysů jako je pozice konečků prstů, excentricita, poměr rozměrů ruky, segmentace pixelů a rotace. Pro extrakci rysů je použit nový algoritmus, který v základě kombinuje algoritmy K curvature a convex hull. Dohromady tato metoda může být na-



Obrázek 5.13: Data získaná z neuronové sítě, a. Modrá čára vyznačuje referenční data použitá při trénování sítě, b. Červená čára vyznačuje výsledky hodnot obrázků s ideálním osvětlením, c. Zelená čára vyznačuje výsledky hodnot obrázků se špatným osvětlením [28]

zvána jako metoda "K convex hull". Tato metoda dokáže detekovat konečky prstů s velkou přesností. V tomto systému je dále použita umělá neuronová síť, která s algoritmy backpropagation dokáže klasifikovat 37 znaků americké abecedy a čísel. Vstupem do sítě je vektor obsahující 30 hodnot rysů, které jsou extrahovány z každého obrázku. Celková úspěšnost rozpoznávání tohoto systému v reálném čase je $94,32\%$.[29]

Pro testování znaků je vytvořeno prostředí GUI (*graphical user interface*), ve kterém se zobrazuje text znázorňovaného znaku. GUI je zobrazeno na obrázku 5.14



Obrázek 5.14: Grafické uživatelské rozhraní GUI[29]

Systém je navržen aby rozeznával všechny statické znaky ASL. Systém kombinuje pět algoritmů pro extrakci rysů. Celý systém pracuje ve čtyřech krocích a to získání obrazu, předzpracování, extrakce rysů a rozpoznávání rysů.

Databáze zpracovávaných znaků obsahuje 1850 obrazů zobrazujících 37 znaků. Každý znak má 50 vzorků pořízených od různých lidí. [29]

Předzpracování obrazu

Kvůli šumu v obrazu pořízeném kamerou je nutné obraz předzpracovat. Obraz je upraven na velikost 260 x 260 pixelů. Poté je z formátu RGB konvertován na černou a bílou pomocí

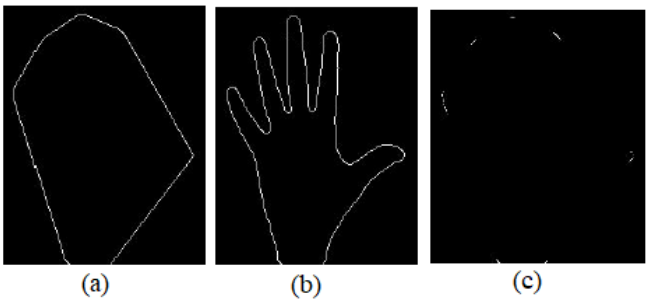
Otsuovy metody. Obraz je dále otočen do pozice kdy zápěstí je dole a pozice prstů míří nahoru. [29]

Klasifikace rysů

Klasifikace rysů je prováděna vícevrstvou neuronovou sítí. Každý znak je charakterizován vektorem obsahujícím 30 hodnot rysů. Neuronová síť má tři vrstvy, vstupní, skrytou a výstupní. Skrytá vrstva má 20 neuronů a výstupní vrstva 37, pro každý klasifikovaný znak jeden. Při testování bylo zpracováno 50 vzorků obrazů pro každý znak. V testovací fázi vykazovala síť přesnost $99,7\%$.[29]

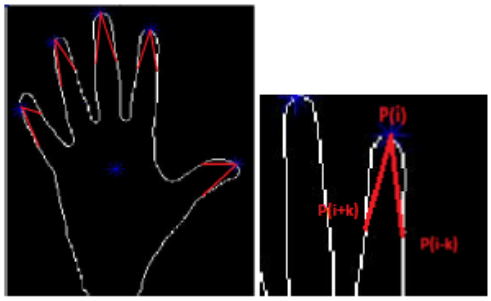
Algoritmus na vyhledávání konečků prstů

Algoritmus pro vyhledávání konečků prstů je realizován za pomoci metod K curvature a Convex hull. Convex hull vytvoří polygon kolem segmentované oblasti ruky, který ohradí všechny body této oblasti. Obrys polygonu je detekován algoritmem hranové detekce Sobel. Poté je provedena operace AND, která najde společné body mezi vytvořeným polygonem a původním obrysem ruky. To razantně sníží čas na zpracování, protože není potřeba používat algoritmus K curvature na všechny pixely obrysu ruky, ale pouze na společné body. Tím je i snížena pravděpodobnost na detekování chybného konečku prstu. Společné body tak jsou jedinými oblastmi ve kterých se nalézají konečky prstů.[29] Výsledky jsou zobrazeny na obrázku 5.15.



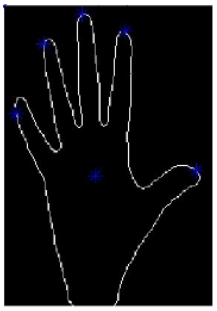
Obrázek 5.15: (a) polygon, (b) originální tvar ruky, (c) společné body[25]

Algoritmus K curvature najde úhly mezi úsečkami $[p(i-k), p(i)]$ a $[p(i), p(i+k)]$, kde p je specifický bod obrysu, i je číslo sekvence specifického bodu, k je vzdálenost od specifického bodu. Hodnota k je nastavena na vzdálenost 40 pixelů. Prahová hodnota úhlu p je nastavena na 50 stupňů. Pokud je tento úhel větší než prahová hodnota, bod p je považován za koneček prstu.[29]



Obrázek 5.16: Aplikace algoritmu K curvature[29]

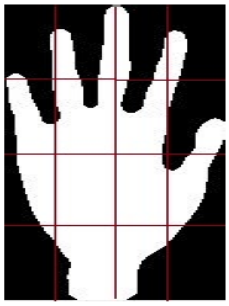
V této metodě je algoritmus K curvature aplikován pouze na společné pixely obrázků po operaci AND. Pixely jsou zobrazeny na obrázku 5.15 (c). Ve chvíli kdy je detekován koneček prstu, je předpoklad že další koneček prstu se nachází v minimální vzdálenosti od tohoto prstu. Je nastaveno, že euklidovská vzdálenost mězi dvěma konečky prstu je 22 pixelů. Tento kombinovaný algoritmus pojmenovaný "K convex hull"vypočítává pozice konečků prstů s větší přesností. Pro extrakci rysů je vypočítána vzdálenost mezi konečky prstů a těžištěm ruky, úhel mezi úsečkami vzniklými spojením těžiště s konečky prstů a horizontální přímkou procházející těžištěm. Dále je jako rys vypočítána celková plocha ruky. Celkem se takto získá 11 rysů, které se posléze klasifikují.[29] Výsledek detekce konečků prstů a těžiště ruky je zobrazen na obrázku 5.17



Obrázek 5.17: Detekce konečků prstů a těžiště ruky [29]

Pixelová segmentace

V algoritmu pixelové segmentace se obrázek rozdělí na šestnáct bloků. V každém bloku se vypočítá počet bílých pixelů. Výsledkem je vektor čítající 16 rysů, každý obsahující počet bílých pixelů v každém bloku.[29] Segmentovaná ruka je zobrazena na obrázku 5.18.



Obrázek 5.18: Pixelová segmentace do 16 bloků [29]

Výsledky

V analýze byl každý znak znázorňován pěti různými lidmi, každý z nich znakoval jeden znak dvakrát. Ve výsledku byl klasifikován každý znak deseti vzorky. Při znakování bylo zachováno černé pozadí a vhodné osvětlení. Získané rysy obrazů byly následně testovány již zmíněnou neuronovou sítí. Průměrná úspěšnost rozpoznávání obrazu navrženého systému byla $94,32\%$. Výsledky dokazují, že algoritmus K convex hull je přesný v detekci konečku prstů a je vhodným prostředkem pro získávání rysů pro klasifikaci znaků.[29]

5.4.3. Postup 3

V této práci je prezentována metoda V. Adithya et al. [30] na rozpoznávání statických znaků Indické prstové abecedy a čísel. Ke klasifikaci je vytvořena databáze obrazů znaků s černým pozadím a vhodným osvětlením. Obraz byl segmentován na bílé pixely označující oblasti ruky a na černé pixely označující pozadí. [30]

Extrakce rysů

Rysy popisující tvar ruky jsou odvozeny ze vzdálenostní transformace černobílého obrazu. Vzdálenostní transformace je odvozené zobrazení obrazu, kde hodnota každého bílého pixelu označujícího objekt je změněna na hodnotu vzdálenosti od nejbližšího černého pixelu označujícího pozadí. Výsledkem je černobílý obraz, kde stupeň šedi koresponduje se vzdáleností nejbližšího hraničního pixelu. V této práci byla použita Euklidovská vzdálenostní transformace, protože je neměnná vůči rotaci obrazu.[30] Euklidovská vzdálenost mezi body $P = (x, y)$ a $Q = (u, v)$ je definována jako:

$$(P, Q) = \sqrt[2]{(x-u)^2 + (y-v)^2} \tag{5.9}$$

V dalším kroku se vypočítá řadový a sloupcový vektor. Každý prvek řadového vektoru R obsahuje sumu nenulových pixelů příslušného řádku v obrazu. Každý prvek sloupcového vektoru C obsahuje sumu nenulových pixelů příslušného sloupce obrazu. Oba vektory jsou 1-D funkce, které reprezentují tvar ruky vstupního obrazu. Tyto tvarové deskriptory zastupující tvar jsou citlivé na šum, takže musí být dále upraveny aby byly robustní.[30]

Fourierovy deskriptory Fourierovy deskriptory transformují koeficienty tvarových deskriptorů popsané výše. Fourierovy deskriptory překonávají citlivost na šum, který je přítomný v tvarových deskriptorech. Mimoto Fourierovy deskriptory zachovávají informace a mohou být jednoduše normalizované.[30] Pro dva vektory R(t)a C(t), kde t=0,1,2...N-1 je diskrétní Fourierova transformace dána vztahem

$$u_n = \frac{1}{N} \sum_{t=0}^{N-1} R(t) \exp\left(\frac{-j\Pi nt}{N}\right) \tag{5.10}$$

kde n=0,1,2...N-1 a N je velikost vektoru R.

$$v_n = \frac{1}{N} \sum_{t=0}^{N-1} C(t) \exp\left(\frac{-j\Pi nt}{N}\right) \tag{5.11}$$

kde $n = 0, 1, 2...N-1$ a N je velikost vektoru R.
Koeficienty u_n a v_n se nazývají Fourierovy deskriptory tvaru.

Vektor rysů Hodnoty rysů jsou vytvořeny z Fourierových deskriptorů řadového a sloupcového vektoru. V potaz se bere pouze velikost Fourierových koeficientů, fázové informace jsou zanedbány. Vektor rysů každého gesta je sestaven ze šesti hodnot a to z druhého, třetího a čtvrtého centrálního momentu normalizovaných Fourierových koeficientů. [30]

Centrální momenty jsou sady hodnot, které charakterizují vlastnosti rozdělení pravděpodobnosti. Pro reálnou náhodnou hodnotu X je kátý centrální moment daný vztahem $\mu_k = E[XE(X)^k]$, kde E je střední hodnota. Nultý centrální moment μ_0 je jedna. První

centrální moment μ_1 je nula. Druhý centrální moment μ_2 se nazývá rozptyl *(variance)* a je často označována jako σ^2. Třetí moment μ_3 a čtvrtý moment μ_4 vyjadřují koeficient šikmosti *(skewness)* a koeficient špičatosti *(kurtosis)*.[30]

Odchylka měří rozložení dat ve vzorku. Je to dobrý deskriptor rozdělení pravděpodobnosti náhodné proměnné. Popisuje rozložení čísel okolo hodnoty průměru. Přestože existuje mnoho metod pro vyjádření různých rozdělení, metody založené na momentech jsou preferovány kvůli jejich výpočetní jednoduchosti.[30] Obecně odchylka sady dat s konečnou velikostí N je dána jako

$$\mu_2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{5.12}$$

kde x_i je hodnota vzorku, $i = 1, 2...N$

Šikmost je měření asymetrie rozdělení pravděpodobnosti reálné náhodné proměnné. Může mít kladnou nebo zápornou hodnotu, nebo může být také nedefinovaná. V tomto případě většina hodnot obsahující i medián leží napravo od průměru takže jsou hodnoty šikmosti kladné. Pokud jsou rozmístěny rovnoměrně na obou stranách kolem průměru je hodnota šikmosti nula.[30] Šikmost sady dat s konečnou velikostí N je dána jako

$$\mu_3 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^3 \tag{5.13}$$

kde x_i je hodnota vzorku, $i = 1, 2...N$

Strmost je měření ostrosti vrcholu rozdělení pravděpodobnosti. Podobně jako koncept šikmosti je strmost také deskriptorem tvaru rozdělení pravděpodobnosti.[30] Strmost sady dat s konečnou velikostí N je dána jako

$$\mu_4 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^4 \tag{5.14}$$

kde x_i je hodnota vzorku, $i = 1, 2...N$

Klasifikace

Vektor získaných rysů je použit jako vstupní klasifikátor na rozeznávání znaků. Umělá neuronová síť je použita jako klasifikační nástroj. Síť má jednu vstupní vrstvu, dvě skryté a jednu výstupní vrstvu. Všechny vrstvy jsou mezi sebou plně propojené. Jako algoritmus pro trénování byl použit algoritmus backpropagation. V trénovací fázi je klasifikováno 36 znaků. Trénovací sada obsahuje 360 obrazů. Každý znak je popsán deseti obrazy. V testovací fázi obsahuje datová sada 180 obrazů, každý znak je zastoupen pěti obrazy. Systém byl implementován za pomoci MATLABR2010a s počítačem, který má procesor i3, 2,2 GHz a paměť RAM 4GB.[30]

Výsledky

Systém je implementován za pomoci MATLABR2010a s počítačem, který má procesor i3, 2,2 GHz a paměť RAM 4GB. Míra úspěšné klasifikace je dána jako podíl mezi znaky správně klasifikovanými a celkovým počtem klasifikovaných znaků. Ve výsledku byla v tomto experimentu míra úspěšně klasifikovaných znaků $91,11\%$. V porovnání s ostatními

metodami má tato nízké výpočetní nároky a velmi vysoký podíl úspěšně klasifikovaných znaků. [30]

6. Vlastní zhodnocení

Při porovnání metod z hlediska náročnosti na provedení z dosažených poznatků vychází, že nejméně náročná je metoda detekce pomocí rukavic, ke které je sice nutný potřebný hardware, ale data získaná z měření jsou lépe zpracovatelná. Data získaná z obrazového záznamu se zpracovávají hůře a jsou více citlivá na šum. Metoda používající zařízení Kinect dosahuje přesných výsledků až 84%, což je vzhledem k náročnosti detekce znaků dobrý výsledek. Metody detekce prstové abecedy z obrazového záznamu dosahují přesností přes 90 %, což je dostatečně přesné, nicméně metody se zaměřují pouze na znaky prstové abecedy, takže je podstatně zjednodušená. Pomocí rukavic se v experimentu od Mohammed Waleed Kadou [8] podařilo dosáhnout přesnosti měření 80%. Dá se předpokládat že s dalším vývojem rukavic se tato přesnost zvětší. Vzhledem k uživatelské dostupnosti jsou metody detekce z obrazového záznamu nejpříznivější. K detekci je potřeba pouze kamera s nepříliš velkým rozlišením. Metoda používající zařízení Kinect potřebuje stejně jako rukavice potřebný hardware, což je pro uživatele přítěží i proto, že se zařízení Kinect již neprodává. Nejvíce perspektivní metodou do budoucna se zdá být metoda detekce pomocí obrazového záznamu, jelikož její vývoj nedosáhl svého maxima.

7. Závěr

Cílem této práce bylo popsat a zhodnotit jednotlivé metody detekce znakové řeči. Práce obsahuje popis detekce pomocí rukavic, pomocí zařízení kinect a pomocí obrazového záznamu.

V první kapitole jsou popsány specifikace znakového jazyka. Důležitým zjištěním je, že znakový jazyk je simultánní povahy. Znakované znaky se mezi sebou překrývají. Tento fakt znamená výrazné ztížení vzhledem k detekci znaků.

V kapitole detekce pomocí rukavic byly zhodnoceny výhody využívání rukavic, z nichž největší je, že data získaná při jejich používaní jsou jasná a stručná a přesně vystihují jednotlivé pohyby rukou a prstů na rozdíl od detekce pomocí videa. Na druhou stranu jsou mechanickou přítěží při znakování a uživateli vadí při pohybu. Největší nevýhodou je, že za předpokladu, že člověk by rukavice používal při běžném životě, by musel uživatel mít rukavice vždy u sebe, což je velmi nepraktické.

V další kapitole bylo popsáno využití senzorického zařízení kinect. Kinect se osvědčil jako senzor získávající 3D obraz, nicméně k jeho používání je nutný hardware, který je nepřenosný a nemá tedy praktické využití.

Poslední kapitola je věnována metodám zpracovávající video nebo jiný obrazový záznam. Vzhledem k tomu, že video je v dnešní době možné pořídit kdykoliv, je tato metoda velice dostupná. Nicméně při procesu extrakce rysů a klasifikace znaků je nutný potřebný výpočetní výkon. Získávání rysů z obrazu je obtížné při porovnání s metodami využívající rukavice. V experimentech se podařilo klasifikovat znaky prstové abecedy s úspěšností více než devadesát procent, což se dá považovat za velice přesné. Jako klasifikační metoda se osvědčily neuronové sítě. Metody zpracovávající obraz dokáží detekovat i ostatní znaky znakového jazyka, avšak znaky jsou detekovány z malé množiny znaků. Pro klasifikaci plnohodnotné znakové řeči čítající tisíce odlišných znaků ještě nebyl proveden žádný experiment. V budoucnu se dá očekávat další pokrok této metody co se týče přesnosti v klasifikaci znaků tak i v navýšení objemu klasifikovaných znaků. Výsledkem práce je souhrn metod pro detekci znákového jazyka, cíle práce byly splněny.

8. Seznam použitých zkratek a symbolů

ČZJ	Český znakový jazyk
NMS	nemanuální signály
DZJ	detekce znakového jazyka
SU	strojové učení
IR	infračervená
VGB	visual gesture builder
CNN	konvoluční neuronová síť
RNN	rekurentní neuronová síť
LSTM	long short-term memory
DNN	hluboká neuronová síť
LRN	lokální odezva normalizace
C	konvoluční vrstva
P	poolovací vrstva
F	plně propojená vrstva
FPGA	Field programmable gate arrays
ASL	Americký znakový jazyk
GUI	Graphical user interface
BP	Backpropagation

Bibliografie

1. *World Federation Of The Deaf* [online] [cit. 2019-05-19]. Dostupné z: `https://en.unesco.org/partnerships/non-governmental-organizations/world-federation-deaf`.
2. REDLICH, Karel. Slyšící dítě hluchých rodič. *Bakalářská práce. Praha: FF UK*. 2003.
3. CERMÁK, Frantisek. Jazyk a jazykoveda. *Prazská imaginace, Praha*. 1997.
4. SERVUSOVÁ, Jana. *Kontrastivní lingvistika-českỳ jazyk x českỳ znakovỳ jazyk*. Česká komora tlumočník znakového jazyka, 2008.
5. COOPER, Helen; HOLT, Brian; BOWDEN, Richard. Sign language recognition. In: *Visual Analysis of Humans*. Springer, 2011, s. 539–562.
6. MOTEJZÍKOVÁ, J. *Simultánnost*. Praha, 2006.
7. ONG, Sylvie CW; RANGANATH, Surendra. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. 2005, č. 6, s. 873–891.
8. KADOUS, Mohammed Waleed et al. Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language. In: *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*. 1996, sv. 165.
9. *Webové stránky CyberGlove*. Dostupné také z: `http://www.cyberglovesystems.com/cyberglove-ii/`.
10. *Smart Gloves Turn Sign Language Gestures Into Vocalized Speech*. Dostupné také z: `https://singularityhub.com/2012/09/16/smart-gloves-turn-sign-language-gestures-into-vocalized-speech/%5C#sm.001450aklpntd5t114a2b2w3moyxe`.
11. SMISEK, Jan; JANCOSEK, Michal; PAJDLA, Tomas. 3D with Kinect. In: *Consumer depth cameras for computer vision*. Springer, 2013, s. 3–25.
12. AHMED, Mateen; IDREES, Mujtaba; ABIDEEN, Zain ul; MUMTAZ, Rafia; KHALIQUE, Sana. Deaf talk using 3D animated sign language: A sign language interpreter using Microsoft's kinect v2. In: *2016 SAI Computing Conference (SAI)*. 2016, s. 330–335.
13. MEKALA, Priyanka; GAO, Ying; FAN, Jeffrey; DAVARI, Asad. Real-time sign language recognition based on neural network architecture. In: *2011 IEEE 43rd Southeastern Symposium on System Theory*. 2011, s. 195–199.
14. *Segmentation in Video Data* [online] [cit. 2019-03-10]. Dostupné z: `http://what-when-how.com/introduction-to-video-and-image-processing/segmentation-in-video-data-introduction-to-video-and-image-processing-part-1/`.
15. RAO, G Ananth; KISHORE, PVV. Selfie video based continuous Indian sign language recognition system. *Ain Shams Engineering Journal*. 2018, roč. 9, č. 4, s. 1929–1939.
16. HARTANTO, Rudy; KARTIKASARI, Annisa. Android based real-time static Indonesian sign language recognition system prototype. In: *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*. 2016, s. 1–6.

17. MASOOD, Sarfaraz; SRIVASTAVA, Adhyan; THUWAL, Harish Chandra; AHMAD, Musheer. Real-time sign language gesture (word) recognition from video sequences using CNN and RNN. In: *Intelligent Engineering Informatics*. Springer, 2018, s. 623–632.

18. A Beginner's Guide to Convolutional Neural Networks (CNNs). *Skymind.ai* [online] [cit. 2019-04-16]. Dostupné z: `https://skymind.ai/wiki/convolutional-network`.

19. SZEGEDY, Christian; VANHOUCKE, Vincent; IOFFE, Sergey; SHLENS, Jon; WOJNA, Zbigniew. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, s. 2818–2826.

20. ABADI, Martín et al. Tensorflow: A system for large-scale machine learning. In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 2016, s. 265–283.

21. *Convolutional Neural Networks (CNNs / ConvNets)* [online] [cit. 2019-04-24]. Dostupné z: `http://cs231n.github.io/convolutional-networks/%5C#layerpat`.

22. BENGIO, Yoshua; SIMARD, Patrice; FRASCONI, Paolo et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*. 1994, roč. 5, č. 2, s. 157–166.

23. HAAPALA, Joonas et al. Recurrent neural networks for object detection in video sequences. 2017.

24. HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*. 1997, roč. 9, č. 8, s. 1735–1780.

25. TOSHEV, Alexander; SZEGEDY, Christian. Deeppose: Human pose estimation via deep neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, s. 1653–1660.

26. KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. 2012, s. 1097–1105.

27. *Prstová abeceda z celého světa* [online] [cit. 2019-05-03]. Dostupné z: `http://www.cds-psn.eu/index/prstova-abeceda-z-celeho-sveta`.

28. VARGAS, Lorena P; BARBA, Leiner; TORRES, CO; MATTOS, L. Sign language recognition system using neural network for digital hardware implementation. In: *Journal of Physics: Conference Series*. 2011, sv. 274, s. 012051. Č. 1.

29. ISLAM, Md Mohiminul; SIDDIQUA, Sarah; AFNAN, Jawata. Real time hand gesture recognition using different algorithms based on American sign language. In: *2017 IEEE international conference on imaging, vision & pattern recognition (icIVPR)*. 2017, s. 1–6.

30. ADITHYA, V; VINOD, PR; GOPALAKRISHNAN, Usha. Artificial neural network based method for Indian sign language recognition. In: *2013 IEEE Conference on Information & Communication Technologies*. 2013, s. 1080–1085.